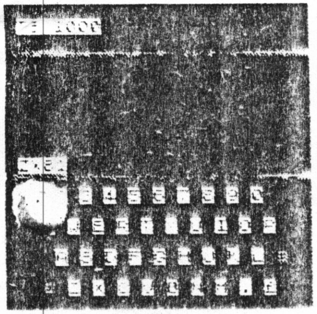


TIMELINEZ

San Francisco/Bay Area Timex/Sinclair USERS

Volume 1 Issue 1 $1.00

Learning Z80 A Assembly Language Programing

In order to program in assembly language we must first look at the " Registers ". Registers are pieces of memory located inside the Microprocessor. They are used to store, retrieve and manipulate numbers, characters, and addresses.

Each type of microprocessor has its own individual set of registers. The names of the registers provided for the z-80A are:

A,B,C,D,E,F,H,L,SP,PC,IX,IY,I,R

The registers A thru L plus I and R are eight bit registers.

Computers talk to themselves in binary arithmetic, in Ø's and 1's. An eight bit register can hold eight Ø's or eight 1's in any desired order. If all the bits in an 8-bit register are set to 1's i.e., 1111111, then the largest number the register can hold is 255 decimal. This is a very limited amount so the 8-bit registers can be paired. When paired, they are now 16-bit registers and can hold a maximum of 65535 decimal or 1111111111111111 binary.

A brief description of the registers and their pairings follows:

"AF" - A is called the accumulator. It is the primary source and destination for most arithmetic operations.

"F" - is the flags or status register. It is an extremly important one. Certain instructions will set or reset these flags as appropriate. F is an 8-bit register with six flags. The names and a brief description of the flags follows:

"s" sign takes on the value of bit 7 of the accumulator after a mathmatical operation.

"z" zero if the result of an instrustion is zero this flag is set

aux carry or half carry=set if a carry was needed from bit 3 to bit 4

P/O = parity/overflow. A dual purpose flag. First it indicates the number of 1's (even or odd) in the 8-bit accumulator after an operation is completed. Second the overflow is set if the sign of a number was changed. More details on this later.

"n" = subtract status set to 1 is subtraction, Ø otherwise.

"c" = carry set if an operation resulted in a number that was larger then the register operated upon could hold. This is not the same as overflow.

"BC" - Historicaly this register is called the byte counter. This register pair, more on pairs later, can be used to count iteration of a program (similar to for-next in basic).

"DE" - sometimes called the destination register. This is just another 16-bit paired register.

"HL" - The primary address pointer. This register is generally used to hold an address. When you want to do something to a specific address it is held in HL. For instance: if you want to load a number into an address you usually load HL with the address then load the contents of HL (the address pointed to by the HL pair) with the number desired. Much more on this later.

"SP" - called stack pointer. This register is used to make a stack. Stacks are very important in programming. One use for the stack is to preserve the values in registers by "pushing" them onto the stack then "popping" them after. One important point to remember about stacks is that first on the stack is the last off. Failure to do this operation will definetly cause a crash.

"PC" - the program counter. Used by the computer to remember where the next instruction to be executed is located. It cannot be manipulated directly by instructions but many of the jump (goto) and call (gosub) instructions manipulate it.

CONT. (2)

Machine Code Graphics: The T/S 1000 (and ZX81) Display File

If you want to program fast (or medium speed) games on the T/S1000, Sinclair BASIC is not appropriate. Z80 machine code is currently the most convenient to use. However, with machine code (MC) you do not have the convenience of the PRINT or PLOT commands. To display characters on the screen you have to POKE the appropriate character code into the proper address of the Display-File (D-File). It is necessary to understand the layout and function of the D-File to be able to use it. The following discussion attempts to describe the D-File of a 16K machine. 1K or 2K D-Files are arranged very differently.

The D-File of a T/S1000 contains the information for the current screen display that you are viewing when your computer is operating. It contains a strip of 793 Bytes in RAM. Each Byte contains either a code for a character to be placed in a specific screen location or a 118 (76 in hex) that tells the machine to end a line. The D-File does not stay in one place in the RAM. However, you can always find it since the current address of the D-File is in a system variable that is stored in addresses 16396 and 16397 (400C and 400D in hex). The D-File starting address is always:

PEEK 16396 + 256*PEEK 16397.

Now let's examine the D-File. Enter and RUN the program in Listing 1:

LISTING 1

```
  10 LET P=PEEK 16396+256*PEEK 1
6397
  20 DIM A(800)
 100 FOR F=0 TO 792
 110 LET A(F+1)=PEEK (P+F)
 120 NEXT F
 200 FOR F=0 TO 792
 210 PRINT A(F+1);" ";
 220 NEXT F
```

The program displays a blank screen while it copies the D-File into the A array. It then prints the D-File of the blank screen onto the screen (the screen will run out of space so use the CONT command to complete the program). As you will see the D-File begins with a 118. There a then 32 0's which mean that the first line is all spaces (see the character set in the appendix of the T/S 1000 Manual). If the top row displayed a row of A's then there would have been 32-38's instead. If you don't believe me add these lines to the program:

```
30 PRINT "AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA"   (32 A's)
130 CLS
```

CONT. (2)

LOCAL USER GROUP MEETINGS

EBZUG	BERKELEY	3RD THURS.
PUG	BURLINGAME	3RD SUN.
SBAZUG	SANTA CLARA	LAST TUES.

CONT. FROM (1)

"IX", "IY" - are index registers. They are used similar to HL , as the usually hold addresses but require an ofset or displacement with them. For instance if you have a table of data and you want the 12th item from it you could load IX with the address of your table plus 12.

"I" and "R" - esoteric registers. I is called the interrupt vector and is used to store the page address of an interrupt response routine. R is called the refresh register and the most refreshing thing about this one is that it is not essential to programming in assembly language.

Binary numbers are extremly hard to program with. The program on the right contains an error can you see it?

00111010	00111010
01100000	01100000
00000000	00000000
01000111	01000111
00111010	01110010
01100001	01100001

To correct this, most assembly language is done in Hexadecimal number system. Hexadecimal is directly transferable to binary. Here is the same program in hexadecimal.

3A	3A
60	60
00	00
47	47
3A	72
61	61

To count in Hex (as it is called) we need to use letters as numbers The number 10 in hex is 16 in decimal . So we must fill the gap between 9 decimal and 10 decimal. Here are the hex digits and their binary equivalents:

01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
0A	1010
0B	1011
0C	1100
0D	1101
0E	1110
0F	1111

In the next artical we will used this infromation to enter a basic program that will help us understand and enter machine code.

F.J.M.
6/3/83

CONT. FROM (1)

After the 0's there is a 118 which tells the machine to end the line. There are then 23 more lines of 33 Bytes (32-0's and 1-118). If any of the 0's were replaced with another character code that character would be printed on the screen.

A handy representation of the D-File is shown Figure 1. It shows each of the screen spaces plus the right hand column which will have all 118's. If you add the number in a specific box to the beginning address of the D-File you can access that screen space. For example, let's put an "inverse space" into space 410 with this program:

```
10 LET P = PEEK 16396 + 256*PEEK 16397
20 POKE P + 410, 128
```

Any character can be placed on the screen using this method. Now add this line and RUN:

```
30 POKE P + 727,23
```

You can see that the 23rd line is now available to us which we cannot access with PRINT and PLOT. The entire screen is available to us in MC programming. Now you might like to try placing different characters to different parts of the screen after defining P as in line 10.

This technique is handy for moving graphics even in BASIC. Figure 1 shows that adding 33 to an address of a space locates the space directly under it. This is important for up and down movement. Diagonal movement is simulated by adding or sbrracting 32 or 34 as in the program in Listing 2:

LISTING 2

```
  10 LET P=PEEK 16396+256*PEEK 1
6397
  20 LET X=-34
  30 LET T=345
 100 POKE P+T,23
 105 LET S=T
 110 LET T=T+X
 120 IF T>725 OR T<1 THEN LET X=
X*(-1)
 130 IF T>725 OR T<1 THEN GOTO 1
10
 140 POKE P+S,0
 150 GOTO 100
```

Changing the value of X in line 20 will cause changes in the movement.

While the examples in this article were in BASIC the principles are necessary for use in MC programming. My article in the May BAZUG 83 newsletter uses this knowledge for a simple MC moving graphic program. That article is reprinted in this issue.

Joel Brody

TIMEX sinclair
ZX 81 T/S 1000
T/S 1500 SPECTRUM T/S 2000
HARDWARE • SOFTWARE • BOOKS
ALL MAJOR BRANDS
SEND 1.00 FOR OUR CATALOG
AVAILABLE JULY 1ST
(REFUNDABLE FIRST ORDER)
SUNSET ELECTRONICS
2254 TARAVAL ST., SAN FRANCISCO, CA J4116
Tel. (415) 665-8330

PROGRAMMING TIPS BY KENDRIC

(1) SAVING BYTES ON YOUR MENUS

LINE 900 (BELOW) IS A COMPLETE MENU. NOTE HOW COMMAS AND SPACES HAVE BEEN USED SO THAT IT WILL PRINT OUT EXACTLY AS THE MENU THAT IS WRITTEN BELOW IN THE MORE GENERAL FORMAT (LINES 900-960). THE BIG DIFFERENCE, HOWEVER, IS THAT LINE 900 USES ONLY 229 BYTES, BUT THE MORE GENERAL FORMAT USES 506 BYTES.

```
 900 PRINT ,,"WHAT IS INCORRECT?
",,"1  MEMBER NO.",,"2  DUES MO.
AND YR.",,"3  SPECIAL CODES","4
FIRST NAMES",,"5  LAST NAME",,"6
  STREET ADDRESS","7  CITY",,"8
 STATE",,"9  ZIP",,"10 PHONE",,"
11 EVERYTHING; DELETE LISTING",,"
12 NOTHING"
```

```
 900 PRINT AT 1,1;"WHAT IS INCOR
RECT?"
 905 PRINT AT 2,1;"1  MEMBER NO.
"
 910 PRINT AT 3,1;"2  DUES MO. A
ND YR."
 915 PRINT AT 4,1;"3  SPECIAL CO
DES"
 920 PRINT AT 5,1;"4  FIRST NAME
S"
 925 PRINT AT 6,1;"5  LAST NAME"
 930 PRINT AT 7,1;"6  STREET ADD
RESS"
 935 PRINT AT 8,1;"7  CITY"
 940 PRINT AT 9,1;"8  STATE"
 945 PRINT AT 10,1;"9  ZIP"
 950 PRINT AT 11,1;"10 PHONE"
 955 PRINT AT 12,1;"11 EVERYTHIN
G; DELETE LISTING"
 960 PRINT AT 13,1;"12 NOTHING"
```

DO YOU HAVE A BETTER WAY TO DO THIS? CALL KENDRIC SMITH AT (415) 493-7210 (PM).

<u>TIMELINZ REVIEWS</u> by David Kinkead

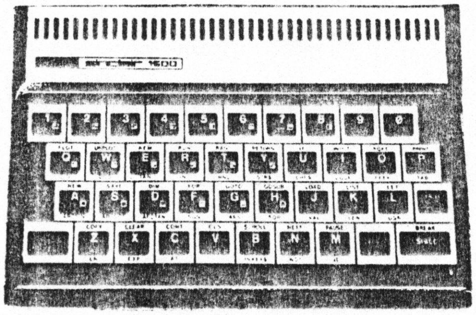
EUREKA! There it was amidst the volumes of cutsie game books and reworded user guides, a volume with the unpretentious name "MASTERING YOUR TIMEX/SINCLAIR 1000 PERSONAL COMPUTER".

"So what's new..." you ask. Tim Hartnell and Dilwyn Jones have done an exceptional job of taking the bewildered newcommer to TIMEX from the basics through some sophisticated programming. Each of its 18 chapters is divided into subchapters and carefully demonstrate the techniques necessary to master the Timex/Sinclair.

The programs provided are very useful (not just asteroid blasting) and contain an explination of why they are listed as they are. It also provides some helpful information on saving space in your programs.

The authors are from Britain. Tim Hartnell is the editor of zx Computing and founder/coordinator for the National ZX Users' Club. Dilwyn Jones runs a users group in North Wales and is a technician in the Welsh broadcasting industry.

Bantam Books, Inc., 666 Fifth Ave., New York, NY 10103. March 1983 $3.95.



COMING SOON ???

- SAME BASIC AND KEYBOARD FUNCTIONS AS TS1000.
- SAME TV MONITOR, BASIC CASSETTE TAPES AND 2040 PRINTER.
- TYPEWRITER STYLE KEYBOARD WITH FULL TRAVEL RUBBER KEYS.
- IMPROVED KEYBOARD LETTERING AND COLORS.
- EXPANDABLE MEMORY - 16K INTERNAL WITH 16K OPTIONAL PLUG-IN MODULE.

IT WILL BE INTEESTING TO SEE HOW TIMEX DECODES MEMORY ADDRESSES FOR BOTH 16K INTERNAL RAM AND 16K EXTERNAL RAM. SOME ADD-ON EQUIPMENT MADE BY OTHER MANUFACTURERS IS PROBABLY IN TROUBLE. ALL CONNECTOR'S HAVE BEEN MOVED TO THE REAR WHICH MAY ALSO BE A PROBLEM FOR SOME ADD-ON EQUIP.

ALL IN ALL, THE NEW 1500 LOOKS LIKE A WINNER.

TIMEX 1000

<u>BUSINESS</u>

Budgeting	$9.95
Inventory Control	$9.95
Financial Analyzer	$9.95
Stock Analyzer	$9.95
Critical Path Analyzer ..	12.95
Bookkeeping	14.95

<u>GAMES</u>

Leap Frog	$9.95
Star Voyage	$9.95
Fungaloids	$9.95
Invaders	$9.95
Chess	$9.95
Snakebite	$9.95
Packman	$9.95

<u>UTILITY</u>

Assembler	$9.95
Disassembler	$9.95
Compiler	$9.95
Graph	$9.95

AND MANY MORE EXCITING PROGRAMS

CALL FOR FREE CATALOG

DISCOUNT SOFTWARE, INC.

320 East 59tn St. New York, N.Y. 10022
Tel:(212)486-0980

TIMELINEZ CENTREFOLD

PUG NEWS

[illegible] THE PENINSULA [illegible] LOOKING FOR- [illegible] WITH THE OTHER [illegible] AREA [illegible] NEWSLETTER. [illegible] SHARING OF INFORM- [illegible] PEOPLE TO FURTHER [illegible] UNDERSTANDING [illegible] COMPUTER.

THE PENINSULA USERS* GROUP [illegible] THE THIRD SUNDAY OF [illegible] AT PENINSULA HOS- [illegible] 1783 EL CAMINO REAL IN BURLINGAME. MEETINGS START AT 1 P.M.

WE ARE FORTUNATE TO HAVE EXCEL- LENT FACILITIES AVAILABLE TO US. SEVERAL ROOMS CAN BE USED FOR DEMONSTRATION OR DISCUSSION PURPOSES AND THERE ARE PLENTY OF TABLES AND ELECTRICAL CON- NECTIONS. THERE IS EVEN A CA- FETERIA FOR SNACKS OR DINNER AFTER THE MEETING.

THANKS GO TO BILL MATTSON FOR FINDING US SUCH A FINE PLACE TO GET TOGETHER. WE HAVE BEEN ASKED BY THE HOSPITAL HOWEVER, NOT TO PARK IN THE VISITOR LOTS. PLEASE USE THE LOTS SUR- ROUNDING THE VISITORS LOT.

ALTHOUGH OUR GROUP IS ONLY A FEW MONTHS OLD WE HAVE SEVERAL PROJECTS IN PROGRESS. FRANK MOURA HAS VOLUNTEERED TO SERVE AS THE PUG EDITOR FOR THE JOINT NEWSLETTER. EVERYONE IS ENCOU- RAGED TO CONTRIBUTE ARTICLES, PROGRAMMING TIPS, REVIEWS ETC. THIS NEWSLETTER CAN ONLY BE AS GOOD AS THE MEMBERS MAKE IT.

WE OPERATE A SOFTWARE LIBRARY IN CONJUNCTION WITH THE SANTA CLARA GROUP. AT PRESENT WE HAVE OVER 60 PROGRAMS AVAILABLE FOR USE BY OUR MEMBERS. (THIS ARTICLE IS BEING WRITTEN ON ONE OF THE PROGRAMS CALLED ZTEXT.) WE ALSO HAVE A WINKY BOARD AVAILABLE TO HELP WITH ANY TAPE LOADING PROBLEMS.

ONE IDEA UNDER CONSIDERATION IS TO BUY SUBSCRIPTIONS TO ALL THE [illegible] PUBLICATIONS AND DONATE THEM TO THE BURLINGAME LIBRARY. [illegible] RESULT IN QUITE A

EBZUG 83

EAST BAY TIMEX/SINCLAIR USERS

WE ARE A NON-PROFIT NEWS- LETTER SERVING USERS IN THE SAN FRANCISCO - OAKLAND BAY AREA.

PRESIDENT JIM MARCH
TREASURER BRUCE BUCHANAN
EDITOR RICK LINK
STAFF JOEL BRODY, WOODY MCPHEETERS, H. POLLARD

PLEASE SEND ALL ARTICLES, ADVERTISEMENTS, AND QUESTIONS BY THE FIRST THURSDAY OF THE MONTH TO: EBZUG
654 40TH ST.
RICHMOND, CA
94805

ALL TRADEMARKS, ETC. ARE PROTECTED. NON-COPYRIGHTED MATERIALS MAY BE USED IN OTHER NEWSLETTERS SO LONG AS CREDIT IS GIVEN TO THE PROPER SOURCE.

NEXT MEETING THURSDAY JUNE 16

MEETING STARTS AT 7:30 PM

WELCOME TO THE FIRST ISSUE OF THE SAN FRANCISCO BAY AREA JOINT NEWSLETTER. MEMBERS OF EBZUG (FORMERLY BAZUG) HAVE CONTRIBUTED ARTICLES AND LABOR TO HELP INSURE THAT THIS NEW VENTURE WILL BE A SUCCESS. EACH USER GROUP WILL CONTINUE IT/S SEPARATE EXISTANCE BUT WILL BENEFIT FROM THE REDUCED PRODUCTION COSTS AND GREATER CIRCULATION THAT THIS NEW PUBLICATION OFFERS. SEE YOU AT OUR NEXT MEETING, JUNE 16 (THE THIRD THURSDAY OF COURSE) AT THE BERKELEY WEST BRANCH LIBRARY NEAR THE CORNER OF UNIVERSITY AND SAN PABLO AVE IN BERKELEY.

SincLink

As the Sinclair ZX has evolved into the TIMEX TS so has SincLink evolved into Timelinez. With this issue we begin a joint effort to pro- duce a new Bay area wide Timex user Newsletter. It's rewarding to see that our efforts to form a true San Francisco Bay Area user group are beginning to bear fruit. This new monthly newsletter is the first ex- ample of increased communication between users that can result from an association of local user groups. In this issue you'll find sections from the North Bay, Peninsula and the South Bay groups.

We welcome and seek the parti- cipation of all Bay area user groups. We want to bring to all users the ideas, questions, answers, exper- iences and expertise that resides in each local group.

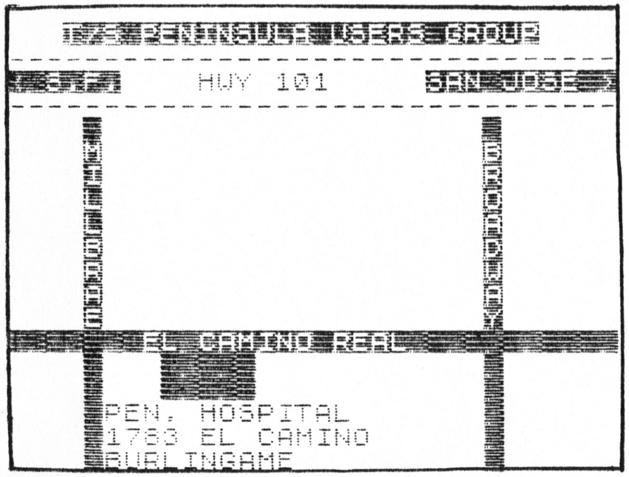
As the amount of hardware and software grows it becomes important to know, before you buy, just what is the wheat so that you don't get the chaff. We hope to use Timelinez to collect and publish your reviews of these new products. Already the North Bay group has made impressive progress in this area.

All readers should recognize and thank George Mockridge for push- ing to make this idea become reality. We also must thank George, Teddy Helderman, Dave Kinkead, Frank Moura and Rick Link for spending their entire Saturday working to put this issue together. Without their effort there would be no newsletter at all!

All readers should also recognize what it takes to continue to put out a quality newsletter. We need your reviews and articles on a piece of hardware or software that you have developed. We'll even help you write them. We also need advertisements

SAVINGS TO OUR MEMBERS AS MORE AND MORE MAGAZINES COME ON THE MARKET, WE WILL DISCUSS THIS IDEA IN MORE DETAIL AT OUR NEXT MEETING.

MANY THANKS TO BOB OKRAFELT FOR HIS FINE SPECTRUM DEMO. AT OUR LAST MEETING. THANKS ALSO TO NEIL STECKLEY FOR GETTING OUT PUBLICITY ABOUT OUR GROUP. AS WE GROW, WE LOOK FORWARD TO MANY ENJOYABLE UNDERTAKINGS.



MEETINGS ARE HELD ON THE 3RD SUNDAY OF EACH MONTH. 1 P.M.

JUNE 19TH
JULY 17TH

NOTE: WE HAVE BEEN ASKED NOT TO PARK IN THE VISITOR LOT. PLEASE USE THE LOTS SUROUNDING THE VISITOR LOT.

ELECTRICAL OUTLETS AND TABLES ARE AVAILABLE SO BRING YOUR EQUIP. AND EXTENSION CORDS IF POSSIBLE.

CAFETERIA SERVICE IS AVAILABLE AND FAMILIES ARE WELCOME.

FOR MORE INFOMATION CONTACT:

GEORGE MOCKRIDGE
263 GATEWAY NO. 107
PACIFICA, CA 94044
(415) 359-3198

SEND NEWSLETTER CONTRBUTIONS TO:

FRANK MOURA
858 CHENERY ST.
SAN FRANCISCO, CA. 94131

NEW MEMBERSHIP PROPOSALS WERE VOTED ON AT THE LAST MEETING, MAY . AS OF THE MAY 19 MEETING THE OFFICIAL EBZUG MEMBERSHIP SCHEDULE IS:

FULL (NEWSLETTER AND LIBRARY PRIVILEGES) $15.00 YEAR
NEWSLETTER ONLY $10.00 YEAR
STUDENT (THRU HIGHSCHOOL) $ 8.00 YEAR

MONTHLY DUES MAY ALSO BE PAID AT EACH MEETING WHICH WILL GUARANTEE TWO NEWSLETTERS BUT DOES NOT INCLUDE LIBRARY PRIVILEGES. HALF YEAR RATES ARE ALSO AVAILABLE ON REQUEST. **NOTE** IF YOU RECEIVED YOUR MEMBERSHIP CARD AT THE LAST MEETING YOU MIGHT HAVE NOTICED THAT THE THERMAL PAPER HAS REACTED WITH THE PLASTIC COVER. WE WILL HAVE NEW CARDS AT THE NEXT MEETING.

NEW BUSINESS FOR OUR NEXT MEETING INCLUDES THE NOMINATION AND ELECTION OF NEW OFFICERS. THIS IS AN IMPORTANT ACTIVITY SO PLEASE BE PRESENT AT OUR NEXT GATHERING.

ROBOTEK
Computer Systems
TOM KILPATRICK
10545 B SAN PABLO
EL CERRITO, CA 94530
(415) 524-3730

TIMELINEZ

THIS IS YOUR NEWSLETTER. IT WILL BE AS GOOD AS YOU MAKE IT. HOW ABOUT CONTRIBUTING AN ARTICLE, REVIEW, LETTER, OR WHATEVER? THANKS.

to help cover the cost of production and mailing. We also need help. No, it won't take a whole Saturday of your time now that we're off and flying.

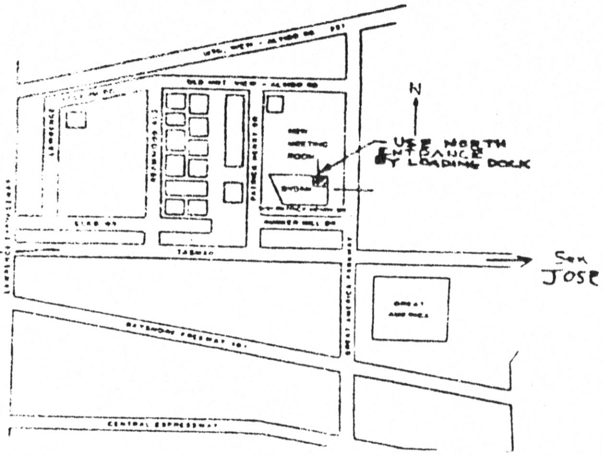
As a result of this new organization our group will now be identified as the South Bay T/S User Group. We will retain the name SincLink for our section of Timelinez.

Our meetings are held on the last Tuesday of each month (not necessarily the 4th Tues.!) between 7 and 10 pm at the Dysan Corp. facilities - see map. At our next meeting Jim McMurry will present our second seminar on machine code programming. He will instruct on how to program in ML and will have several example ML programs available. The meetings are open and all are welcome to attend. See you there!

Best wishes,

Paul

Paul D. Perreault
President



The regular meeting of the ZUG is: 7:00 to 10:00 PM, the LAST TUESDAY of each month.
DYSAN Corporation
5401 Patrick Henry Drive
Santa Clara, CA.
Use the NORTH enterence by the loading dock.

On this page Eric Reiter, our own local TS computer wizzard, brings us up to date on his latest projects.

Now available for TS1000/ZX81 with 16K RAM.

Printer Plotter in high resolution graphics; cassette tape & manual..$35.

Business file management program; cassette tape with manual..........$10.

To be released in June 1983:

Fast Fourier Transform on EPROM. Two 2716s or one 2764..............$75.

Dual trace storage scope with high resolution printer graphics & FFT.$100.

About the programs.

FAST FOURIER TRANSFORM. (256 point, 8 bit precision , magnitude, phase, complex coefficients, ONE SECOND, Hamming window). This program is 4 Kilobytes of solid Z80 machine code specially written to our specifications and high standards. It is used for spectrum analysis and other scientific applications. Here is an example of its use: If you set up a bank on RAM of 256 bytes which represents a waveform of voltage (magnitude) verses time, the FFT program will...

1) grab the 256 bytes of data from the bank,
2) perform a prescaling of the data called a Hamming Window (optional),
3) do the FFT,
4) set up a bank of the real and imaginary coefficients (source of mag and phase),
5) set up a 128 byte bank which can describe a graph of voltage vs frequency.
6) set up a 128 byte bank which can describe a graph of phase shift vs frequency.

The waveform has now been completely described in terms of the magnitude and phase shifts of a set of sine waves. The entire operation takes less than one second on the Timex which uses a 3.25 megaHertz clock. The data delivered is a full 8 bits of precision. The calculations have been performed with 16 bit precision to avoid round offs. The program is available in EPROM at addresses 8192 to 12288 in the 'transparent' part of the Timex memory map. The RAM banks start at 29K and can work in the 16K RAM with or without moving 'ramtop'. The EPROMs are available in either two 2726's or one 2764. The 2716's can fit on the Hunter Board. The 2764 can fit on the UM64 from Byte Back. Documentation will be furnished with the FFT. A complete manual including use, theory and applications of the FFT is in progress and can be delivered at a future date. We also plan to have the FFT available on disk for computers that run CPM.

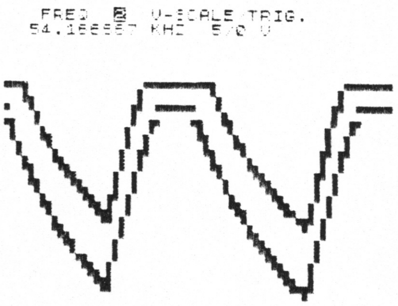
DUAL TRACE STORAGE SCOPE. This is a report on the progress of our "Scope" product. It uses the high resolution Printer Plotter by John Kane. It is designed to work with the Computer Continuum Analog Interface Board, TS/ZX printer, 16K or more of RAM, and a ZX/TS computer. Upon RUNning the program the Scope automatically does a series of data acquisitions and displays the trace on the video monitor or TV. The user enters single stroke commands as outlined in the table below.

command key	numerical entry	function performed
F	.003 to 155	sampling Frequency in kiloHertz.
T	0 to full scale	Trigger voltage setting.
C	1 or 2	Channel select.
S		do a scope Sweep (data acquisition & display).
R		Repeat sweeps every 3 seconds.
P		Printer plot with high resolution graphics.
N		Enter a label or name to be placed on the printer plot and do the printer plot.
V	positive real number	set scale of Voltage full scale.

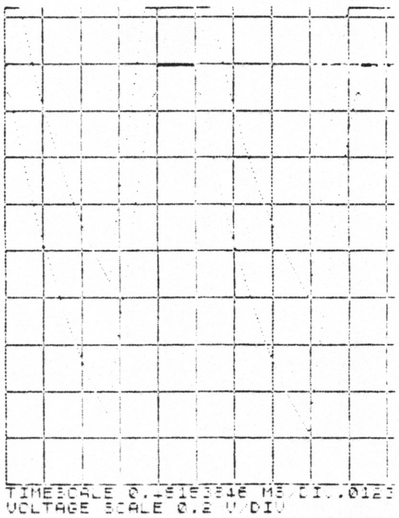
Each sweep takes 1.5 seconds to produce the video display. In single channel mode the sampling frequency range is 3 Hz to 155 kiloHertz. In dual channel mode the sampling frequency range is 3 Hz to 60 kiloHertz. Analog data is retrieved through I/O channels 0 and 1.

In progress is a cursor and numerical readout routine and spectrum analysis routines using the FFT.

COPY to printer of TV screen:

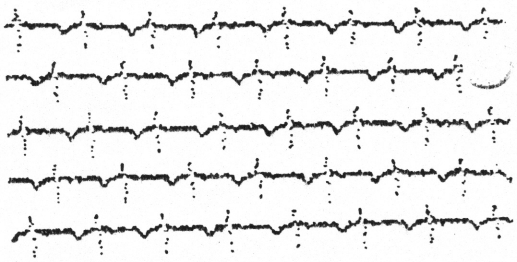


High resolution printer display:

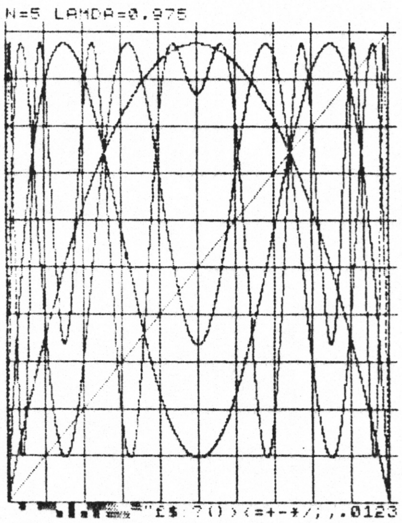


Computer Continuum
301-16th Ave
San Francisco, CA. 94118 (415)752 62??

PRINTER PLOTTER. This program can be used as a subroutine to plot in pixel graphics on the ZX or Timex printer. It is written in machine code and BASIC. It is well documented and modular. The program is arranged in 4 parts. Part 1 can be used to clear the plot image in memory. Part 2 can draw a 10 X 10 grid (graticule) in memory. Part 3 is used to plot a point at position X,Y. Part 4 is used to generate or acquire data to be plotted. Data may be fed to the printer-plotter with either BASIC or machine code. A subroutine is included which is used to acquire data through an I/O port address. The time interval between successive I/O data transfers is a variable. After each time interval the X value will be incremented to the right one pixel. The number of dots plotted per time interval is a variable, thus plots with steep grades can be filled in for easier visual analysis. The number of readings per dot is a variable and is used to average the data. These features are especially useful when the Plotter program is used with our Analog Interface Board. This program has been used with a heart monitor and the Analog Board to produce the display *below*.



This program was written by J. V. Kane and Co, 109 Ardmore Av, Ardmore PA 19003. J. V. Kane and Co will be most pleased to answer any inquires you may have about this program.



N=5 LAMDA=0.975

ISMOVE: A Simple Moving Graphic in Machine Code

When you get your first computer and learn a few BASIC commands one of the first things you do is make something move. A new T/S 1000 user may produce a program like this:

```
10 LET X=0
20 PRINT AT 8,X;"■"
30 PRINT AT 8,X;" "
40 LET X=X+1
45 IF X=32 THEN STOP
50 GOTO 20
```

It's quite exciting to see the moving "inverse space" (IS) at first, but after a while you realize how slow it moves and that Sinclair BASIC has very limited facilities for speeding it up. The following Z80 machine code program works the same way as the above BASIC program does. However it is executed so fast that a DELAY subroutine had to be added so your eye and display screen can deal with it. Its great speed also enables us to add a speed control facility in an accompanying BASIC program.

ISMOVE is 34 Bytes long. If you don't use an assembler or have a favorite machine code loading program type in the following program.

```
1 REM ..................................
10 FOR F=16514 TO 16547
20 INPUT A
30 POKE F,A
40 NEXT F
```

Now type in the following numbers starting with 42 (16514 is the address of the first Byte). This is the decimal machine code listing:

```
16514
42  12  64  35  1   42  1   9
6   31  62  128 119 197 205 158
64  205 158 64  193 62  0   119
35  16  239 201 6   102 0   16
253 201
```

Those who prefer to work in hexidecimal can instead use the following listing:

HEXADECIMAL LISTING

HEX ADDR.	HEX M.C.	HEX ADDR.	HEX M.C.
4082	2A 0C 40	409E	C1
4085	23	409F	3E 00
4086	01 2A 01	4099	77
4089	09	409A	23
408A	06 1F	409B	10 EF
408C	3E 80	409D	C9
408E	77	409E	06 FE
408F	C5	40A0	00
4090	CD 9E 40	40A1	10 FD
4093	CD 9E 40	40A3	C9

The Z80 mnemonic listing and explanatory notes are as follows (all numbers are hexadecimal except those preceded by a "+"):

****

Z80 MNEMONIC LISTING

```
START LD HL,(400C)   ;SEE NOTE A
      INC HL
      LD BC,+298
      ADD HL,BC
      LD B,+31        ;SEE NOTE B
LOOP  LD A,80         ;SEE NOTE C
      LD (HL),A
      PUSH BC         ;SEE NOTE D
      CALL DELAY      ;SEE NOTE E
      CALL DELAY
      POP BC
      LD A,0          ;SEE NOTE F
      LD (HL),A
      INC HL
      DJNZ LOOP       ;SEE NOTE G
      RET

DELAY LD B,+254       ;SEE NOTE H
DLOOP NOP
      DJNZ DLOOP
      RET
```

Notes:

A) The first 4 lines locates the Display File and adds 298 to the address. This will locate the first space in the 8th line. (It is important to understand the layout of the Display file for doing M.C. graphics.)
B) B is the counter for the DJNZ instruction (see F).
C) 80 is the hexadecimal character code for the IS. This line loads it into its proper place in the Display File.

D) This instruction saves the counter by putting it on the stack so it can be retrieved later with the POP instruction. We need the BC register in the DELAY subroutine.
E) The IS will move too fast unless we put in the DELAY subroutine. CALL is the M.C equivalent of GOSUB. 2 CALLs are included to give greater control of speed.
F) A "space" is substituted for the IS.
G) Return to LOOP and decrement B if B>0.
H) NOP is "no operation". It is an empty loop that cycles the number of times as the number that is loaded in B. We can regulate the speed of the moving IS by POKEing in a different value for B

****

After you have entered the machine code type in:

```
10 RAND USR 16514
```

and delete lines 20, 30 and 40. You should see the IS move steadily across the screen if you have entered the program properly. The speed of the IS is faster than the BASIC program, but it is still rather slow. However, as mentioned in note B we can regulate the speed by changing the value in register B in the DELAY subroutine. The address of this number is 16543. The speed can be changed by POKEing a number between 0 and 255 into this address (0 is the fastest). The following program will help you do this (do not change your current lines 1 and 10):

```
1 REM [illegible]
10 RAND USR 16514
20 PRINT "DO YOU WANT TO CHANGE SPEED? (Y OR N)"
30 INPUT N$
40 IF N$="Y" THEN GOTO 100
45 CLS
47 PRINT "SPEED FACTOR= ";PEEK 16543
50 RAND USR 16514
55 INPUT A$
57 IF A$="C" THEN GOTO 20
60 GOTO 50
100 CLS
105 PRINT "INPUT SPEED FACTOR 0 TO 255"
108 PRINT "0 IS FAST, 255 IS SLOW"
110 INPUT A
115 PRINT "SPEED FACTOR= ";A
120 POKE 16543,A
130 GOTO 50
```

RUN this program and it will ask you if you want to INPUT a new "speed factor". After you do this the program will run each time you press N/L. Pressing "C" instead of N/L will let you change the "speed factor" again. You will find that you will not even be able to see the moving IS at low values of the "speed factor"

A further excercise would be to put the moving IS on a different line and to try to make the IS move in different directions.

JOEL SPIC

A MYSTERY PROGRAM

The following "MYSTERY PROGRAM" is by a Toll Holt and was first published in the February 1982 issue of a British magazine called "YOUR COMPUTER". Here is the program with a slight variation:

```
1 PRINT "HERE IS A 10 YEAR BINARY COUNTER"
2 LET A=PEEK 16396+256*PEEK 16397+32
3 LET B=PEEK A=157
4 POKE A,157-B
5 LET A=A-1
6 GOTO 2+B
```

The mystery is to discover what it does and how it works. Type it in your computer and RUN it before reading further.

Now that you're reading further, I'll explain the "mystery" to me. The first line just reserves 32 spaces (for 1K or 2K computers). Line two sets A to the value on the left of the first line of the display file. Line 3 looks like a mistake!! However the program works. Are you setting A to 157 and PEEKing at ROM address 157? ALL WRONG! The next line POKEs A so A can't be 157. After some more confusion I found line 3 reads as follows: LET B = either a 1 if the number at address A is 157 (true) or LET B = a 0 if the number is other than 157 (false). The first equal sign tells the computer what B is (assignment) and the second equal sign tells the computer to test for true or false (boolean). So B is either 1 or 0.

Line 4 pokes an inverse 1 if B is 0 or an inverse 0 if B is 1.

Line 5 sets the address one place to the left.

Line 6 loops back to line 2 if PEEK A was an inverse 0 or to line 3 if an inverse 1.

I hope all this is now clear. The next mystery is for you to solve. How do you save this program?

Bob Orrfelt

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	64	55	56	57	58	59	60	61	62	63	64	65	66
67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132
133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198
199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264
265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297
298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330
331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363
364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396
397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429
430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462
463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495
496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528
529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561
562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594
595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627
628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660
661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693
694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726
727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759
760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792

FIGURE 1

THIS COLUMN CONTAINS ALL 118'S

HIGHFALUTIN' COMPUTIN'

with

Bob Orrfelt

on your

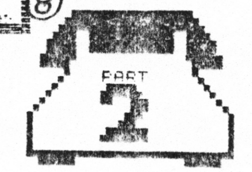
TS 1000 Computer

Intro to advanced ideas.
Dice to hex and more.
120 pages, 30 programs

For your copy send
$11.60 inc s&h to

Bob Orrfelt
3436 Bay Road,
Redwood City, Ca 94063

TIMELINEZ ⑧



TELEPHONE LIST

LOAD "FONE"

FEATURES:

Search Routine: alpha-numeric, finds full name or first letter of names, or finds a phone number by the last 4 digits. (i.e. that forgotten long distance call on your phone bill) A sorting sub-routine saves search time by moving more frequently searched items to the top of the list. A floating end-point saves time by not searchińg the unused section of the list.

MENU DRIVEN COMANDS: LIST...(prints out all list items)
ADD....(allows additions to list)
DELETE.(removes from list)
SEARCH.(finds name/number)
RECALL.(finds number/name)
FILE...(files ammended list)

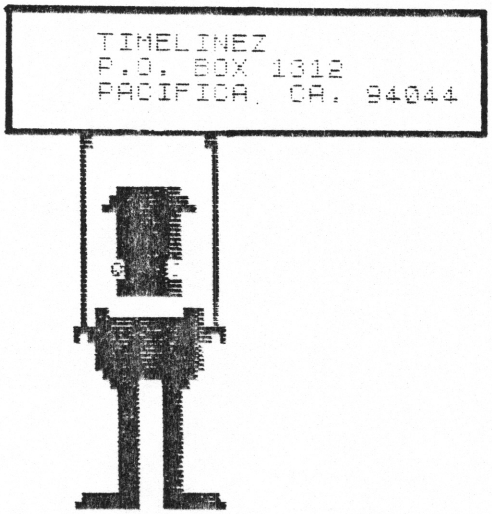
SET-UP ROUTINE: Takes only a minute, 1 time.
Allows you to name the list.
Allows you to set the size of list.
Clears a used list.

THE SET-UP COMMANDS ARE:

```
L        ENTER
-space-     ENTER
RUN 2    ENTER
LET B$="name of list"      ENTER
LET A=size of list      ENTER
95     ENTER
GOTO  30     ENTER
```

THESE COMMANDS ARE ENTERED WITHOUT LINE NUMBERS

Use GOTO 1 ENTER, after a -break- or an error code, to return to the Menu. On occasion you might get an error code 5/line#, this happens when your search returns more lines than the screen can hold. C ENTER will allow you to continue. You can input 0,1, or up to 18 letters to search, i.e. SMITH will return all Smith names, including Smithe. If you're not sure of the spelling, Search SM and get Smith, Smithe, and Smythe John.

NOTES: This program was written for 16K, TIMEX 1000, and will hold 300+ names/numbers in 16K. If you have 32K memory you can LET A=700 and the list can hold 700 names, with a corresponding increase in LOAD time. I recommend a "size of list" of 100 as a convenient size as this fits nicely on a C-5 tape. B$ is any name you wish to head the list, it can be up to 14 characters long. Line 95 is a list of set up commands in case you lose this sheet. L ENTER after you LOAD the first time, prints out information stored in array; it is removed by GOTO 30 when you set up.

```
SET-UP ROUTINE
ENTER THESE COMMANDS-NO LINE NO.

L
SPACE
RUN 2
LET B$="14 MAX"
LET A=50
95
GOTO 30

   1 GOTO 60
  10 REM COPYRIGHT 1983 GERALD P
AXTON
  20 DIM B$(14)
  30 DIM N$(A,32)
  40 LET B=0
  50 LET E$="
                 "
  60 CLS
  70 PRINT B$;" TELEPHONE LIST"
  80 PRINT
  90 PRINT "LIST ALL NAMES/NUMBE
RS";TAB 27;"KEY L";"ADD NAME/NUM
BER";TAB 27;"KEY A";"DELETE NAME
/NUMBER";TAB 27;"KEY D",,"SEARCH
 FOR NAME";TAB 27;"KEY S";"RECAL
L NUMBER";TAB 27;"KEY R",,"TO FI
LE ON TAPE";TAB 27;"KEY F"
  95 PRINT ,,,,"SET-UP ROUTINE",
,"ENTER THESE COMMANDS-NO LINE N
O.",,"L",,"SPACE",,"RUN 2",,"LET
 B$=""14 MAX""",,"LET A=50",,"95
",,"GOTO 30"
 100 IF INKEY$="" THEN GOTO 100
 110 IF INKEY$="L" THEN GOTO 180
 120 IF INKEY$="A" THEN GOTO 340
 130 IF INKEY$="D" THEN GOTO 620
 140 IF INKEY$="S" THEN GOTO 910
 150 IF INKEY$="R" THEN GOTO 115
0
 160 IF INKEY$="F" THEN GOTO 129
0
 170 GOTO 100
 180 CLS
 190 LET X=0
 200 LET J=0
 210 LET X=X+1
 220 LET J=J+1
 230 PRINT N$(X)
 240 IF J=20 THEN GOSUB 290
 250 IF X<B THEN GOTO 210
 260 PRINT "LAST ITEM ON LIST"
 270 GOSUB 290
 280 GOTO 80
 290 PRINT
 300 PRINT "KEY ""ENTER"" TO CON
TINUE"
 310 INPUT Z$
 320 CLS
 330 RETURN
 340 CLS
 350 LET B=B+1
 360 LET X=1
 370 IF B>A OR X>A THEN GOTO 410
 380 IF N$(X)=E$ THEN GOTO 490
 390 LET X=X+1
 400 GOTO 370
 410 PRINT "LIST FULL";
 420 LET B=B-1
 430 FOR J=1 TO 23
 440 PRINT "*";
 450 NEXT J
 460 PRINT
 470 PRINT
 480 GOTO 70
 490 CLS
 500 LET B=X
 510 PRINT "NAME ?"
 520 INPUT N$(X, TO 18)
 530 PRINT "AREA CODE ?"
 540 INPUT N$(X,20 TO 22)
 550 PRINT "FIRST 3 DIGITS OF PH
ONE NO.?"
 560 INPUT N$(X,25 TO 27)
 570 PRINT "LAST 4 DIGITS ?"
 580 INPUT N$(X,29 TO 32)
 590 PRINT N$(X)
 600 PAUSE 150
 610 GOTO 60
 620 CLS
 630 PRINT "DELETE NAME ?"
 640 INPUT Z$
 650 LET X=1
 660 IF N$(X, TO LEN Z$)=Z$ THEN
GOTO 700
 670 IF X=B THEN GOTO 880
 680 LET X=X+1
 690 GOTO 660
 700 CLS
 710 PRINT N$(X)
 720 PRINT
 730 PRINT "TO DELETE";TAB 15;"K
EY ""D"",""ENTER"""
 740 PRINT
 750 PRINT "FOR NEXT """;Z$;""""
 760 PRINT "KEY ""ENTER"""
 770 INPUT Z$
 780 CLS
 790 IF Z$<>"D" THEN GOTO 670
 800 PRINT ,,N$(X),"DELETED"
 810 LET N$(X)=E$
 820 FOR X=X TO B-1
 830 LET N$(X)=N$(X+1)
 840 NEXT X
 850 LET N$(B)=E$
 860 LET B=B-1
 870 GOTO 890
 880 PRINT "LAST ITEM"
 890 PAUSE 100
 900 GOTO 60
 910 CLS
 920 PRINT "NAME ?"
 930 INPUT Z$
 940 LET F=0
 950 LET X=1
 960 CLS
 970 PRINT "SEARCHING...";Z$
 980 IF N$(X, TO LEN Z$)<>Z$ THE
N GOTO 1030
 990 LET F=1
1000 LET Y=X
1010 LET T$=N$(X)
1020 PRINT T$
1030 LET X=X+1
1040 IF X<=B THEN GOTO 980
1050 PRINT
1060 IF F=0 THEN PRINT " """;Z$
""" NOT FOUND"
1070 IF F=1 THEN PRINT "NO MORE
""";Z$;""""
1080 PRINT "ON LIST"
1090 IF F<>1 THEN GOTO 1140
1100 FOR X=Y TO 2 STEP -1
1110 LET N$(X)=N$(X-1)
1120 NEXT X
1130 LET N$(1)=T$
1140 GOTO 80
1150 CLS
1160 PRINT "LAST 4 DIGITS OF NUM
BER ?"
1170 INPUT Z$
1180 LET F=0
1190 LET X=1
1200 CLS
1210 PRINT "SEARCHING...";Z$
1220 IF N$(X,29 TO 32)<>Z$ THEN
GOTO 1250
1230 LET F=1
1240 PRINT N$(X)
1250 LET X=X+1
1260 IF X<=B THEN GOTO 1220
1270 IF F=0 THEN PRINT Z$;" NOT
LISTED"
1280 GOTO 70
1290 CLS
1300 PRINT AT 10,10;"START TAPE"
,TAB 10;"KEY ""ENTER"""
1310 INPUT Z$
1320 SAVE "FON█"
1330 GOTO 1
```